

Sprint 1 Functions:

Name	createNewAccount
Job	Create a new row for a new user in Accounts database
Parameters	username: String email: String password: String type: int
Return	nothing

Name	credentialsMatch
Job	Check if the password that user input is right
Parameters	idtype: "username" or "email", indicates the type of id. id: String, could be an email or a username, depends on idtype. password: String
Return	true, if the credentials match false o/w null, if the username does not exist

Name	usernameForEmail
Job	Return the username with the given email
Parameters	email: String, the email to be searched for
Return	the username of user with email = email, if the user is found null o/w

Name	updateCredentials
Job	Update user's email/password with the new one
Parameters	type: String, either 'email' or 'password' username: String, the user who requested the update newCredential: String, the new email or password, depending on type
Return	nothing

Name	addToken
Job	Create a new row for a new user in Token database
Parameters	token: String, the user's token username: String expirationTime: Number, int
Return	nothing

Name	deleteToken
Job	Delete user's token in the database
Parameters	token: String, the token to be deleted.
Return	nothing

Name	getUsernameByToken
Job	Return the username with the given token
Parameters	token: String, the token to be deleted.
Return	username: String, the username corresponding to the token. null if the token is not found in the database, or has expired.

Sprint 2 Functions:

Name	addUserProfile
Job	Create a new row for a new user in Profile database
Parameters	username: String
Return	nothing

Name	updateProfile
Job	Update user's profile in the Profile database
Parameters	username: String, the user updates: - updates: Object, keys are fields to be updated, values are the new value for the said field
Return	nothing

Name	getProfileByUserName
Job	Return a JavaScript Object where each field contains user's profile info in the database
Parameters	username: String
Return	An object. Keys are each field. Values are the data in the database.(use 'undefined' or 'NaN' to check if the data exist)

Name	getUserByType
Job	Return the type of the user
Parameters	username: String
Return	A string indicating user's type

Name	createUser
Job	Create a user node in the database
Parameters	username: String
Return	nothing

Name	makePost
Job	Create a postNode with given data and create 'CREATE_POST' relationship between user and post
Parameters	username: String content: String, the content of the post title: String, the title of the post postId: the unique postId of the post time: the time that user makes the post
Return	nothing

Name	deletePost
Job	Delete the post node and any relationship with this post node (However, related reply will not be deleted)
Parameters	postId: the postId of the post that user reply to
Return	nothing

Name	searchPostByTitle
Job	Return a set of objects where each object contains postId, title and time
Parameters	title: String, part of the string of the title
Return	A set of object looks like: [{postId: A01, title: new_world, time: 2000-01-01, content: "Hi"}, {postId: B09, title: hello_world, time: 2020-10-10, content: "Hello"}]

Name	getPosts
Job	Return a set of objects where each object contains postId, title and time sorted by time, and returns posts numbers skip = pageNum * postPerPage limit = postPerPage
Parameters	pageNum: int postsPerPage:int
Return	A set of object looks like: [{postId: A01, title: new_world, time: 2000-01-01, content: "Hi"}, {postId: B09, title: hello_world, time: 2020-10-10, content: "Hello"}]

Name	getComments
Job	Return a set of objects that are comments of the original post where each object contains postId, title and time sorted by time
Parameters	postId: String
Return	An object filled with comments under the original post

Name	searchPostById
Job	Return a set of objects where each object contains postId, title and time
Parameters	Id: Part of the target postId
Return	A set of object where each object contains postId, title, time and content (similar to the above method)

Name	searchPostByUser
Job	Return a set of objects where each object contains postId, title and time. Each object will be the post that user created
Parameters	Username: the username of the user
Return	A set of object where each object contains postId, title, time and content (similar to the above method)

Name	makeReply
Job	Create a reply node, 'CREATE_REPLY' relation between user and the reply and 'REPLY_TO' relation between the post/reply and the reply
Parameters	username: String content: String, the content of the Reply replyId: the unique id of the reply targetId: the Id of the target that user reply to time: the time that user makes the reply type: the type of the target (should be "post" or "reply")
Return	nothing

Name	deleteReply
Job	Delete the reply, 'CREATE_REPLY' and any 'REPLY_TO' relation with this reply
Parameters	replyId: the unique Id of the reply
Return	nothing

Name	addTag
Job	Create a tag and the 'HAS_TAG' relationship between the user and tag. If the tag exists in the database (has the same tagName), then only the "HAS_TAG" relationship will be created.
Parameters	username: String tagName: String
Return	nothing

Name	getTags
Job	Gets all the tags that a user has.
Parameters	username: String
Return	An object containing all of the tags that the user has.

Name	userHasTag
Job	Checks if the user has the given tag
Parameters	username: String tagName: String
Return	True if the user has a 'HAS_TAG' relationship with the tag, False if the user does not.

Name	deleteTag
Job	Delete the relationship between the user and the tag, and the tag node if there are no relationships involving the tag
Parameters	username: String tagName: String
Return	nothing

Name	followPost
Job	Create 'FOLLOW' relationship between the user and the post
Parameters	username: String postId: String, the postId of the post that user wants to follow
Return	nothing

Name	unfollowPost
Job	Delete 'FOLLOW' relationship between the user and the post
Parameters	username: String postId: String, the postId of the post that user wants to unfollow
Return	nothing

Name	getFollowedPostByUser
Job	Return a set of postId where each post is followed by the user
Parameters	username: String
Return	A set of postId that user follows Empty if user follows nothing

Name	getFollowingUserByPost
Job	Return a set of username where each user is following this post
Parameters	postId
Return	A set of username where each user is following this post Empty if user follows nothing

Sprint 3 Functions:

Name	getModule
Job	Return a set that contains all the content in the module
Parameters	moduleId
Return	A set of username where each user is following this post Empty if user follows nothing

Name	createModule
Job	Create a module of a course and a relationship between the course and the module
Parameters	course the name of the course id the module id name the name of the id
Return	Nothing

Name	addContentIntoModule
Job	Create HAS_CONTENT relationship with an object
Parameters	type type of object (deliverable, video or reading) objId the id of the object moduleId the id of the module
Return	Nothing

Name	deleteAllContent
Job	Delete all the relationship of content with a module
Parameters	moduleId the module id
Return	Nothing

Name	getAllDeliverable
Job	Return a set of object that contains all the deliverable in the module
Parameters	moduleId the id of the module
Return	a set of object that each object contains deliverable title, media, content and due

Name	getAllVideo
Job	Return a set of object that contains all the video in the module
Parameters	moduleId the id of the module
Return	a set of object that each object contains video name, description and url

Name	getAllReading
Job	Return a set of object that contains all the reading in the module
Parameters	moduleId the id of the module
Return	a set of object that each object contains reading name, description and path

Name	getAllModules
Job	Get all the modules and their content of a course
Parameters	course the name of the course
Return	a set where each object contains the name of the module and its content

Name	createDeliverable
Job	Store the deliverable info in the database. If the deliverable does not have due, it will be set to -1
Parameters	name String description String deliverableId String timestamp String moduleId String
Return	A set of username where each user is following this post

	Empty if user follows nothing
--	-------------------------------

Name	updateCourse
Job	Edit the course description
Parameters	name the name of the course description optional. the new description instructor optional. username of the new instructor
Return	Nothing

Name	deleteModule
Job	Delete the module and any relationship with the module
Parameters	id the id of the module
Return	Nothing

Name	searchModuleById
Job	Return an object that contains the feature of the module
Parameters	id the id of the module
Return	an object that contains the course, id and name of a module

Name	checkIsInstructor
Job	Check whether the user is an instructor of the course
Parameters	moduleId the id of the module instructor the username of the module
Return	true if the user is an instructor false if not

Name	getCourses
Job	Return all the course that user enrolls in
Parameters	username String
Return	A set of course id

Name	searchCourseByName
Job	Search the course by its name and return an object containing its feature
Parameters	courseName String
Return	A set of course where each object contains its feature

Name	createCourse
Job	Create the course in the database and its relation to the instructor
Parameters	courseName String instructor String description String
Return	Nothing

Name	deleteCourse
Job	Delete the course
Parameters	courseName String
Return	Nothing

Name	getCourseContent
Job	Return the content of the course
Parameters	courseName String

Return	an object that contains course info and modules
--------	---

Sprint 4 Functions:

Name	createVideo
Job	Create the video in the database
Parameters	id the id of the video name the name of the video description the description of the video url the url of the video posted_timestamp the posted time of the video
Return	Nothing

Name	editVideo
Job	Edit the description and url for the video
Parameters	id the id of the video description the new description url the new url
Return	Nothing

Name	deleteVideo
Job	Delete the video
Parameters	id the id of the video
Return	Nothing

Name	searchVideoById
Job	Return a object that contains the feature of the video
Parameters	id the id of the video
Return	an object that contains the id, name, description, url and post-time

Name	createReading
------	---------------

Job	Create the reading in the database
Parameters	id the id of the Reading name the name of the Reading description the description of the Reading path the path of the file posted_timestamp the posted time of the Reading
Return	Nothing

Name	editReading
Job	Edit the name and content of the reading
Parameters	id the id of the Reading name the name of the Reading description the description of the Reading
Return	Nothing

Name	deleteReading
Job	Delete the reading in the database
Parameters	id the id of the Reading
Return	Nothing

Name	searchReadingById
Job	Return a object that contains the feature of the reading
Parameters	id the id of the Reading
Return	an object that contains the id, name, description, path and post-time

Name	createSubmission
Job	Create a submission, a CREATE_SUBMISSION relationship with the user and a SUBMIT_TO relationship with the deliverable
Parameters	username the username of the user

	deliverableId the id of the deliverable id the unique id of the submission content the content of the submission media the submission media posted_timestamp the date when the submission is submitted
Return	Nothing

Name	gradeSubmission
Job	Set the grade of the submission to the new grade. If missing the second paramaters, it will be set to 0
Parameters	id the id of the submission comment the comments for the submission grade the grade of the submission
Return	Nothing

Name	searchSubmissionById
Job	Get the submission
Parameters	id the submission id
Return	the username, content, media, posted time, and grade of the submission

Name	searchCourses
Job	Get the course with given criteria
Parameters	username (optional) when provided, this function will also return whether the user is enrolled in the course or not. criteria can be an object with any combination of the following fields <pre> { 'name_contains': string, 'enrolled_by': string, 'name_equals': string, 'has_module': string, 'has_content': string }</pre>
Return	a set of course with the criteria